UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/007,498	11/13/2001	Hung T. Nguyen	01-625	2278

24319 7590 07/22/2004

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106 LEGAL
MILPITAS, CA 95035

EXAMINER	
RAVINDRAN, LATHA	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

8

Office Action Summary	**Application No.** 10/007,498	**Applicant(s)** NGUYEN ET AL.	
	Examiner Latha Ravindran	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _13 November 2001_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-20_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-20_ is/are rejected.

7)☒ Claim(s) _1,8, 15_ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _13 November 2001_ is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) □ Notice of Informal Patent Application (PTO-152)
6) □ Other: _____.

DETAILED ACTION

1. Claims 1 – 20 have been examined. Claims 1 – 20 have been rejected.

Drawings

2. Figure 1 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g).

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: Fig. 2. The ISU is coupled to the DEU/ETM. There is no mention of the DEU/ETM in the description of the drawing.

4. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the

- Claim 6: “wherein grouping logic within said processor groups said multiply-accumulate instructions based on said mechanism”
- Claim 9, 12: “interim results are unavailable to an external program executing in said processor”
- Claim 13: “grouping said multiply-accumulate instructions based on said mechanism”
- Claim 19: “interim results are unavailable to an external program executing in said DSP”
- Claim 20:”grouping logic groups said multiply-accumulate instructions based on said mechanism”

must be shown or the features canceled from the claims. No new matter should be entered.

5. The drawings are objected to:

- Remove the word “Figure” and replace with “Fig.” in Figures 1 – 6. See 37 CFR 1.84 (u)(1).
- Replace the handwritten notation in Figs. 1- 6 with its typed equivalent in order to create clean, reproducible characters in the drawings. See 37 CFR 1.84(l).
- The examiner requests that DSP 100 be labeled as DSP 100, not just 100, for clarity.

6. Corrected drawing sheets are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled “Replacement Sheet” in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

7. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required:

- Claim 4: "...wherein said out-of-order completion logic writes back said interim results to at least one register in said MAC before multiply-accumulate instructions arrive at said accumulation stage of said MAC..."
- Claim 11: "...writing back said interim results to at least one register in said MAC..."
- Claim 18: "...wherein said out-of-order completion logic writes back said interim results to at least one register in said MAC..."

In the specification, the interim results are written back to the ORF. (Page 23, Paragraph 50, " In the illustrated embodiment, ...", Page 25, Paragraph 55, "The final and leftover sums and carries..."), not MAC.

8. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

9. Revise the first sentence of the abstract so it is in narrative form and does not contain the form and legal phraseology often used in patent claims. See MPEP 608.01(b).

10. The section headings should be in upper case, without underlining or bold type. Please remove the bold from the headings. See 37 CFR 1.77 and MPEP 608.01(a).

11. The specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

12. Claims 1,8, and 15 are objected under MPEP 2173.05(e) for being indefinite due to a lack of antecedent basis in the claim.

- Claim 1: "...allows younger instructions..."
- Claim 8: "...allowing younger instructions..."
- Claim 15: "...allows younger instructions..."

The term, "younger instructions" is indefinite. Do younger instructions refer to younger multiply accumulate instructions or younger instructions in general? Only multiply accumulate instructions have been named in the claim.

13. Upon further examination, the term "younger instructions" refer to instructions that follow older instructions. (Page 5, Paragraph 10)

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 1- 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Motorola, Inc. (MPC7410 RISC Microprocessor Technical Summary) and in further view of Morris (Computer Architecture The Anatomy of Modern Processors).

Claim 1:

16. Motorola discloses:

17. For use in a processor (MPC7410, Page 3, Figure 1),

- having an at least four-wide instruction issue architecture (Page 4, “As many as four instructions can be fetched from the instruction cache per clock cycle.”/ The four-wide instruction issue architecture is illustrated in the capability to fetch up to four instructions from the instruction cache in a clock cycle.)
- a mechanism for pipeline processing multiply-accumulate instructions (Page 3, Figure 1, Floating Point Unit, Page 2, The paragraph beginning with “The FPU and VFPU are pipelined; ...”, The paragraph beginning with “Note that for the MPC7410...”/ The mechanism for pipeline processing multiply-

accumulate instructions is the FPU. fmadds is a multiply-add instruction, which is a type of multiply-accumulate instruction.)

- with out of order completion (Page 3, Figure 1, Page 31 – 33, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / After the instructions are dispatched in order, they execute in variable latency execution units, and enter the completion unit out of order. Then they are retired in order.)
- comprising:
- a multiply-accumulate unit (MAC) having an initial multiply stage and a subsequent accumulate stage; (Page 32, Figure 6, FPU1 – FPU3, Page 33, "The FPU stages are multiply, add, and round-convert."/ The FPU, floating point unit, is the MAC unit. The accumulate stage is the add stage.)
- out of order completion logic (Page 3, Figure 1, Page 31 – 33, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / The processor has out of order completion logic. The Dispatch Unit, Independent Execution Units (AltiVec Vector Permute Unit, AltiVec Vector Arithmetic Logic Unit, Integers Units, Floating-Point Unit) ,VR File 6 Rename Buffers, GPR File 6 Rename Buffers, FPR File 6 Rename Buffers, and Completion Unit comprise the out of order completion logic.)
- associated with said MAC (Page 3, Figure 1,Dispatch Unit, VR File 6 Rename Buffers, GPR File 6 Rename Buffers, FPR File 6 Rename Buffers, Floating Point Unit, Completion Unit / The Floating Point Unit is coupled to the Dispatch Unit, Rename Buffers, and Completion Unit),

- allows younger instructions to complete before said multiply-accumulate instructions. (Page 31 – 36, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / After the instructions are dispatched in order, they execute in variable latency execution units, and enter the completion unit out of order. The floating point multiply/add is a type of multiply-accumulate instruction. Younger instructions whose execution units have less latency than the floating point unit (such as the integer unit) can complete before the floating-point multiply/add instruction. This is the out of order completion. Then the instructions are retired in order.)

18. However, Motorola is silent about:

- [out of order logic, associated with said MAC,] that causes interim results produced by said multiply stage to be stored when said accumulate stage is unavailable

19. Motorola teaches that the floating point unit, which executes the multiply-add instruction, is pipelined. (Page 2, The paragraph beginning with, "The FPU and VFPU are pipelined;..." and "Note that for the MPC 7410, ..." Page 33, "The FPU stages are multiply, add, and round-convert."). Motorola teaches out of order logic, associated with said MAC. (Page 3, Figure 1, Page 31 – 33, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / After the instructions are dispatched in order, they execute in variable latency execution units, and enter the completion unit out of order. Then they are retired in order. The floating point unit, or MAC, is coupled to the Dispatch Unit, Rename Buffers, and Completion Unit). Motorola is silent about [out of order logic, associated with said MAC,] that

causes interim results produced by said multiply stage to be stored when said accumulate stage is unavailable.

20. Morris teaches that results of each stage of a pipelined processor are latched in registers as an input for the following stage. (Morris, Page 1, Second Paragraph). These registers are referred to as pipeline registers.

21. The desired advantage to latching the results of each stage, and thus employing a pipelining technique, is to improve throughput, and speed, of the processor. (Morris, Page 1)

22. Referring to Motorola's MPC7410, the results of each stage of the pipelined processor are stored in pipeline registers described by Morris. Motorola's floating-point unit has three stages, multiply, add, and round-convert. (Motorola, Page 33). The pipeline registers of the floating-point unit are a part of the out of order logic, associated with said MAC. The results of the multiply stage of the floating-point unit (the interim results produced by the multiply stage) are stored in a pipeline register before the add stage can process them (to be stored when said accumulate stage is unavailable). Motorola's processor has the results stored for use by the next pipeline stage of the floating-point unit. Motorola's processor now has the desired advantage of increased throughput and speed.

23. One of ordinary skill in the art at the time of applicant's invention would have been motivated to store interim results in a pipeline register of a pipelined processor to achieve increased throughput and speed. The pipeline registers of floating point unit (with the multiply, add, and round-convert stages) are a component of out of order logic, associated with said MAC, that causes interim

results from the multiply stage to be stored when said accumulate stage is unavailable.

24. Therefore, it would have been obvious to combine the pipeline register described by Morris to Motorola's processor to obtain the invention as specified in claim 1.

Claim 2:

25. The mechanism as recited in Claim 1, wherein said initial multiply stage and said subsequent accumulate stage are single clock cycle stages. (Motorola, Page 2, the paragraphs beginning with "The FPU and VFPU are pipelined; ..." "Note that for the MPC7410, ...")

Claim 3:

26. The mechanism as recited in Claim 1 wherein said out-of-order completion logic is contained in a writeback stage of a pipeline in said processor. (Motorola, Page 32, Figure 6, Page 33, the paragraph beginning with, "The complete pipeline stage ...."/ The completion unit is in the writeback stage because it retires instructions.)

Claim 4:

27. The mechanism as recited in Claim 1 wherein said out-of-order completion logic writes back interim results to at least one register in said MAC before said multiply-accumulate instructions arrive at said accumulation stage of said MAC. (Motorola, Page 32, Figure 6, FPU1 – FPU 3, Page 33, "The FPU stages are multiply, add, and round-convert." Page 2, "For example, a floating-point multiply-add instruction takes three cycles to execute, regardless of whether it is single-

(**fmadds**) or double-precision. (**fmadd**)." / The floating-point unit is also the MAC. In the floating point unit, the multiply stage writes its results, the interim results, in a pipeline register, described by Morris in Claim 1, before the add stage further processes the instruction.)

Claim 5:

28. The mechanism as recited in Claim 1 wherein said interim results are unavailable to an external program executing in said processor. (Motorola, Page 21, Figure 5 lists all accessible registers. /The pipeline registers, which hold the interim results, are not accessible to the programmer. Note that other internal storage, such as rename buffers, and reservation stations, are unavailable to the external program.)

Claim 6:

29. The mechanism as recited in Claim 1 wherein grouping logic within said processor groups said multiply-accumulate instructions based on said mechanism. (Motorola, Page 3, Figure 1,Page 8, 2.2.1 Instruction Queue and Dispatch Unit, Page 25, 3.2.1 PowerPC Instruction Set /The grouping logic is the dispatch unit. It sends floating point instructions, including the floating point multiply-add instruction, to the Floating Point Unit, which is also the MAC unit.)

Claim 7:

30. The mechanism as recited in Claim 1 wherein said processor is a digital signal processor. (Motorola, Page 36, AltiVec technology/ The MPC7410 is a processor designed for digital signal processing.)

Claim 8:

31. Motorola discloses:

32. For use in a processor (MPC7410, Page 3, Figure 1)

- having an at least four-wide instruction issue architecture (Page 4, "As many as four instructions can be fetched from the instruction cache per clock cycle."/ The four-wide instruction issue architecture is illustrated in the capability to fetch up to four instructions from the instruction cache in a clock cycle.)
- a method of pipeline processing multiply-accumulate instructions(Page 3, Figure 1, Floating Point Unit, Page 2, The paragraph beginning with "The FPU and VFPU are pipelined; ...", The paragraph beginning with "Note that for the MPC7410..."/ The method of pipeline processing multiply-accumulate instructions is illustrated by the execution of fmadds and fmadd instructions through the floating-point unit. (FPU). fmadds and fmadd are floating-point multiply-add instructions, which are a type of multiply-accumulate instruction.)
- with out-of-order completion, (Page 3, Figure 1, Page 31 – 33, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / After the instructions are dispatched in order, they execute in variable latency execution units, and enter the completion unit out of order. This is the out of order completion. Then they are retired in order.),
- comprising:
- providing a multiply-accumulate unit (MAC) having an initial multiply-stage and a subsequent accumulate stage; (Page 32, Figure 6, FPU1 – FPU3,

Page 33, "The FPU stages are multiply, add, and round-convert."/ The FPU, floating point unit, is the MAC unit. The accumulate stage is the add stage.)

- allowing younger instructions to complete before said multiply-accumulate instructions. (Page 31 – 36, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / After the instructions are dispatched in order, they execute in variable latency execution units, and enter the completion unit out of order. The floating-point multiply-add is a type of instruction. Younger instruction whose execution units have less latency than the floating point unit (such as the integer unit) can complete before the floating point multiply/add instruction. This is the out of order completion. Then they are retired in order.)

33. However, Motorola is silent about:

- causing interim results produced by said multiply stage to be stored when said accumulate stage is unavailable;

34. Motorola teaches that the floating point unit, which executes the multiply-add instruction, is pipelined. (Page 2, The paragraph beginning with, "The FPU and VFPU are pipelined;..." and "Note that for the MPC 7410, ..." Page 33, "The FPU stages are multiply, add, and round-convert.") Motorola is silent about causing interim results produced by said multiply stage to be stored when said accumulate stage is unavailable.

35. Morris teaches that results of each stage of a pipelined processor are latched in registers as an input for the following stage. (Morris, Page 1, Second Paragraph). These registers are referred to as pipeline registers.

36. The desired advantage to latching the results of each stage, and thus employing a pipelining technique, is to improve throughput, and thus speed, of the processor. (Morris, Page 1)

37. Referring to Motorola's MPC7410, the results of each stage of the pipelined processor are stored in pipeline registers described by Morris. Motorola's floating-point unit has three stages, multiply, add, and round-convert. (Motorola, Page 33). The results of the multiply stage of the floating-point unit (the interim results produced by the multiply stage) are stored in a pipeline register before the add stage can process them (to be stored when said accumulate stage is unavailable). Motorola's processor has the results stored for use by the next pipeline stage of the floating-point unit. Motorola's processor now has the desired advantage of increased throughput and speed.

38. One of ordinary skill in the art at the time of applicant's invention would have been motivated to store interim results in a pipeline register of a pipelined processor to achieve higher throughput and speed. The interim results produced by the multiply stage are stored in a pipeline register when the add stage is unavailable.

39. Therefore, it would have been obvious to combine the pipeline register described by Morris to Motorola's processor to obtain the invention as specified in claim 8.

Claim 9:

40. The method as recited in Claim 8 wherein said initial multiply stage and said subsequent accumulate stage are single clock cycle stages. (Motorola,

Page 2, the paragraphs beginning with "The FPU and VFPU are pipelined; ..." "Note that for the MPC7410, ...")

Claim 10:

41. The method as recited in Claim 8 wherein said causing is carried out in a writeback stage of a pipeline in said processor. (Page 32, last paragraph/ Even though the pipeline registers are in the execute stage, the instruction may depend on data from instructions in the writeback stage. The result from the instruction in the writeback stage, found in the reorder buffer, then causes the floating point multiply/add instruction to execute, and thus use the pipeline register to store interim results stored in the multiply stage.)

Claim 11:

42. The method as recited in Claim 8 wherein said causing comprises writing back said interim results to at least one register in said MAC before said multiply-accumulate instructions arrive at said accumulation stage of said MAC. (Motorola, Page 32, Figure 6, FPU1 – FPU 3, Page 33, "The FPU stages are multiply, add, and round-convert." Page 2, "For example, a floating-point multiply-add instruction takes three cycles to execute, regardless of whether it is single- (**fmadds**) or double-precision. (**fmadd**)." / The floating-point unit is also the MAC. In the floating point unit, the multiply stage writes its results, the interim results, in a pipeline register, described by Morris in Claim 8, before the add stage further processes the instruction.)

Claim 12:

43. The method as recited in Claim 8 wherein said interim results are unavailable to an external program executing in said processor. (Motorola, Page 21, Figure 5 lists all accessible registers. /The pipeline registers, which hold the interim results, are not accessible to the programmer. Note that other internal storage, such as rename buffers, and reservation stations, are unavailable to the external program.)

Claim 13:

44. The method as recited in Claim 8 further comprising grouping said multiply-accumulate instructions based on said mechanism. (Motorola, Page 3, Figure 1,Page 8, 2.2.1 Instruction Queue and Dispatch Unit, Page 25, 3.2.1 PowerPC Instruction Set /The grouping logic is the dispatch unit. The dispatch unit sends floating point instructions, including the floating point multiply-add instruction, to the Floating Point Unit, which is also the MAC unit.)

Claim 14:

45. The method as recited in Claim 8 wherein said processor is a digital signal processor. (Motorola, Page 36, AltiVec technology/ The MPC7410 is a processor designed for digital signal processing.)

Claim 15:

46. Motorola discloses:

47. A digital signal processor (DSP), comprising:

- a pipeline having stages (Page 31, 3.6 Instruction Timing, The paragraph beginning with “The MCP7410 is a pipelined, superscalar processor. ...”

- and capable of processing multiply-accumulate instructions; (Page 25, 3.2.1 PowerPC Instruction Set/ Floating-point multiply/add instructions)
- an instruction issue unit containing grouping logic (Page 3, Figure 1, Dispatch Unit, Page 2.2.1 Instruction Queue and Dispatch Unit)/ The Dispatch Unit, the instruction issue unit, contains grouping logic.)
- and at least four-wide instruction issue logic; (Page 3, Figure 1, Page 4, "As many as four instructions can be fetched from the instruction cache per clock cycle."/ The four-wide instruction issue logic is fetcher in Figure 1. It fetches up to four instruction from the 32 K-byte I cache.)
- a multiply-accumulate unit (MAC), (Page 32, Figure 6, FPU1 – FPU 3, Page 33, first paragraph/ The floating point unit (FPU) is the MAC unit.)
- coupled to said instruction issue logic, (Page 3, Figure 1, Dispatch Unit, Floating Point Unit)
- having an initial multiply stage and a subsequent accumulate stage; and (Page 33, "The FPU stages are multiply, add, and round-convert."/ The initial multiply stage is the multiply stage. The subsequent accumulate stage is the add stage.)
- out of order completion logic, (Page 3, Figure 1/ The processor has out of order completion logic. The Dispatch Unit, Independent Execution Units (AltiVec Vector Permute Unit, AltiVec Vector Arithmetic Logic Unit, Integers Units, Floating-Point Unit) ,VR File 6 Rename Buffers, GPR File 6 Rename Buffers, FPR File 6 Rename Buffers, and Completion Unit comprise the out of order completion logic).

- associated with said pipeline, (Page 3, Figure 1, Page 32, Figure 6, The paragraph beginning with "The instruction pipeline in the MPC7410 has four major pipeline stages, described as follows:..."/ The components of the out of order completion logic are all located in the pipeline of the processor.)
- that allows younger instructions to complete before said multiply-accumulate instructions. (Page 31 – 36, 3.6 Instruction Timing, Page 9, 2.2.3 Completion Unit / After the instructions are dispatched in order, they execute in variable latency execution units, and enter the completion unit out of order. This is the out of order completion. The floating-point multiply-add is a type of instruction. Younger instructions whose execution units have less latency than the floating point unit (such as the integer unit) can complete before the floating-point multiply/add instruction. Then they are retired in order.)

48. Motorola is silent about:

- [out-of-order] completion logic that causes interim results produced by said multiply stage to be stored when said accumulate stage is unavailable

49. Motorola teaches that the floating point unit, which executes the multiply-add instruction, is pipelined. (Page 2, The paragraph beginning with, "The FPU and VFPU are pipelined;..." and "Note that for the MPC 7410, ..." Page 33, "The FPU stages are multiply, add, and round-convert."). Motorola also teaches out-of-order completion logic, associated with said pipeline. (Page 3, Figure 1, Page 32, Figure 6, The paragraph beginning with "The instruction pipeline in the MPC7410 has four major pipeline stages, described as follows:..."/ The processor has out of order completion logic. The Dispatch Unit, Independent Execution Units

(AltiVec Vector Permute Unit, AltiVec Vector Arithmetic Logic Unit, Integers Units, Floating-Point Unit) VR File 6 Rename Buffers, GPR File 6 Rename Buffers, FPR File 6 Rename Buffers, and Completion Unit comprise the out of order completion logic. The components of the out of order completion logic are all located in the pipeline of the processor.) Motorola is silent about [out-of-order completion logic, associated with said pipeline,] that causes interim results produced by said multiply stage to be stored when said accumulate stage is unavailable.

50. Morris teaches that results of each stage of a pipelined processor are latched in registers as an input for the following stage. (Morris, Page 1, Second Paragraph). These registers are referred to as pipeline registers.

51. The desired advantage to latching the results of each stage, and thus employing a pipelining technique, is to improve throughput and speed of the processor. (Morris, Page 1)

52. Referring to Motorola's MPC7410, the results of each stage of the pipelined processor are stored in pipeline registers described by Morris. Motorola's floating-point unit has three stages, multiply, add, and round-convert. (Motorola, Page 33). The floating-point unit, and their associated pipeline registers, are a part of the out of order logic, associated with said pipeline. The results of the multiply stage of the floating-point unit (the interim results produced by the multiply stage) are stored in a pipeline register before the add stage can process them (to be stored when said accumulate stage is unavailable). Motorola's processor has the results stored for use by the next pipeline stage of

the floating-point unit. Motorola's processor now has the desired advantage of increased throughput and speed.

53. One of ordinary skill in the art at the time of applicant's invention would have been motivated to store interim results in a pipeline register of a pipelined processor to achieve increased throughput and speed. The pipeline registers of floating point unit (with the multiply, add, and round-convert stages) are a component of out of order logic, associated with said MAC, that causes interim results from the multiply stage to be stored when said accumulate stage is unavailable.

54. Therefore, it would have been obvious to combine the pipeline register described by Morris to Motorola's processor to obtain the invention as specified in claim 15.

Claim 16:

55. The DSP as recited in Claim 15 wherein said initial multiply stage and said subsequent accumulate stage are single clock cycle stages. (Motorola, Page 2, the paragraphs beginning with "The FPU and VFPU are pipelined; ..." "Note that for the MPC7410, ...")

Claim 17:

56. The DSP as recited in Claim 15 wherein said out-of-order completion logic is contained in a writeback stage of said pipeline. (Motorola, Page 32, Figure 6, Page 33, the paragraph beginning with, "The complete pipeline stage ...."/ The completion unit is in the writeback stage of the pipeline because it retires instructions.)

Claim 18:

57. The DSP as recited in Claim 15 wherein said out-of-order completion logic writes back said interim results to at least one register in said MAC before said multiply-accumulate instructions arrive at said accumulation stage of said MAC. (Motorola, Page 32, Figure 6, FPU1 – FPU 3, Page 33, "The FPU stages are multiply, add, and round-convert." Page 2, "For example, a floating-point multiply-add instruction takes three cycles to execute, regardless of whether it is single- (**fmadds**) or double-precision. (**fmadd**)." / The floating-point unit is also the MAC. In the floating point unit, the multiply stage writes its results, the interim results, in a pipeline register, described by Morris in Claim 15, before the add stage further processes the instruction.)

Claim 19:

58. The DSP as recited in Claim 15 wherein said interim results are unavailable to an external program executing in said DSP. (Motorola, Page 21, Figure 5 lists all accessible registers. /The pipeline registers, which hold the interim results, are not accessible to the programmer. Note that other internal storage, such as reservation stations and reorder buffers, are unavailable to the external program.)

Claim 20:

59. The DSP as recited in Claim 15 wherein said grouping logic groups said multiply-accumulate instructions based on said mechanism. (Motorola, Page 3, Figure 1,Page 8, 2.2.1 Instruction Queue and Dispatch Unit, Page 25, 3.2.1 PowerPC Instruction Set /The dispatch unit is the grouping logic. It sends floating

point instructions, including the floating point multiply-add instruction, to the Floating Point Unit, which is also the MAC unit.)

Conclusion

60. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Ruetz et al. European Patent Application 0517241A2, 6 May 1992
  - Figure 2 – conventional double pipelined multiplier-accumulator
  - This multiplier accumulator has two stages, an initial multiply stage and a subsequent accumulate stage.
  - The pipeline register between the multiply stage and the accumulate stage is the storage for interim results produced by the multiply stage when the accumulate stage is unavailable.
- LSI Logic Corporation, An Overview of the ZSP Architecture White Paper, 2000.
  - Describes a superscalar that dispatches four instructions per cycle
- Wang et al. US Pat. 5,826,055 10 October 2000

  Retires groups of instructions executed in or out of order

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Latha Ravindran whose telephone number is

(703)305-8115. The examiner can normally be reached on Monday through Friday 8:30am to 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Latha Ravindran
Latha Ravindran
Examiner
Art Unit 2183

Eddie C
EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100